



UNIVERSIDAD TÉCNICA "LUIS VARGAS TORRES"
DE ESMERALDAS

INVESTIGACIÓN FORMATIVA
PROYECTO DE AULA

TEMA:
Diseño y Desarrollo de una Plataforma Web para Gestión de la
Información Académica

Docente tutor: Ing. Stalin Francis Quinde.
Cursos Participates: 5to A Ingeniería de Software I
5to B Ingeniería de Software I

ASIGNATURA :
Ingeniería de Software I

PERIODO:
2023-2S

Índice

1. Datos Generales	3
2. Introducción	3
3. Objetivo del Proyecto	4
3.1. Objetivo General	4
3.2. Objetivos Específicos	4
3.3. Actividades del proyecto de desarrollo de software (Ciclo de vida del software)	4
3.4. Diagrama de Gantt	6
4. Actividades de aprendizaje basado en Desarrollo del Proyecto	8
4.1. Actividades de configuración y aprendizaje base para el inicio del proyecto	8
4.2. Levantamiento de requerimientos	10
4.2.1. Actividad A1-1	10
4.3. Análisis y Diseño	12
4.4. Actividad A1-2	12
4.4.1. Actividad B1-1	13
4.4.2. Actividad B1-2	14
4.5. Actividad C1-1	15
4.6. Actividad C1-2	16
4.7. Desarrollo del software aplicando la metodología Ágil - Scrum	17
4.7.1. Actividad E1	17
4.8. Análisis y diseño	18
4.8.1. Actividad A2-1	18
4.9. Despliegue	19
4.9.1. Actividad A2-2	19
4.10. Documentación, mantenimiento y difusión	20
4.10.1. Actividad B2-1	20
4.10.2. Actividad B2-2	21
4.10.3. Actividad C2-1	22
4.10.4. Actividad C2-2	23
4.11. Actividad E2	24
5. Resultados	25
6. Conclusiones	25

Índice de figuras

1.	Actividad A1-1:Taller para elaboración de Historias de usuario	10
2.	Actividad a1-2:Taller para la elaboración de prototipos	12
3.	Actividad B1-1:Validación de historias de usuario y prototipos con usuario final del sistema.	13
4.	Actividad B1-2: Ver, escuchar y comentar los vídeos de las sesiones clase	14
5.	Actividad C1-1:Lectura de artículo científico	15
6.	Actividad C1-2: Lectura de artículo científico	16
7.	Dramatización Actividad E1	17
8.	Actividad A2-1: Documento Especificaciones de Requerimiento	18
9.	Actividad A2-2: Despliegue de módulo	19
10.	Diagrama entidad relación	20
11.	Diagrama de Clases	20
12.	Actividad B2-2: Explicación de modulo desplegado	21
13.	Actividad C2-1: Aprendizaje autónomo - Lectura de artículo científico.	22
14.	Actividad C2-2: Aprendizaje autónomo Lectura de artículo científico.	23
15.	Actividad E2:Presentación final del producto del Proyecto de desarrollo de Software .	24

1. Datos Generales

Facultad:	Ingenierías
Carrera:	Tecnología de la Información
Asignatura:	Ingeniería de Software I
Nivel:	5to
Numero de estudiantes:	50
Número de profesoras:	1
Tiempo de ejecución:	1 ciclo (4 meses)
Periodo académico:	2024-1S
Fecha de Inicio:	1 de Agosto del 2024
Fecha de culminación:	1 - Diciembre-2024
Número de varones:	30
Número de mujeres:	20
Inversión del proyecto	Auto-financiado por el docente y los estudiantes
Localización geográfica	Campus Nuevos Horizontes
Producto de proyecto de Aula:	Versión mejorada del Sistemas de Información SICA
Línea de investigación:	La transformación y uso de la materia, fuerza, energía y sistemas informáticos
Sublínea de investigación:	La tecnología de la información y comunicación como aporte al desarrollo de la ciencia en el entorno de la era digital
Evaluación:	Publicación de artículo en una revista indexada de segundo cuartil o grupo 2.

2. Introducción

En el ámbito educativo actual de la Universidad Técnica Luis Vargas Torres de Esmeraldas, la gestión eficiente de datos se tiene que convertir en un elemento crucial para el desarrollo y funcionamiento efectivo de esta institución educativa. La necesidad de acceder, organizar y utilizar la información de manera sistemática esta motivando a docentes y estudiantes a trabajar de forma colaborativa para la creación de herramientas tecnológicas personalizadas que facilitan estas tareas. En este contexto, el docente que dicta la asignatura de Ingeniería de software 1, utiliza una metodología de aprendizaje basada en proyecto, la cual se enfoca en el "Diseño y Desarrollo de un Software de Gestión de Datos para una Institución Educativa-[tapia2020pedag](#)<empty citation>.

La complejidad de la administración académica y operativa de una institución educativa demanda una solución integral que abarque desde la gestión de estudiantes y personal hasta la coordinación de actividades y recursos. Este proyecto tiene como objetivo principal el desarrollo de un software robusto y eficiente que permita a la institución optimizar sus procesos internos, mejorando la toma de decisiones y proporcionando una experiencia más fluida tanto para el personal académico, administrativo así como a los estudiantes que trabajen en el proyecto.

Durante el transcurso de este proyecto, se llevará a cabo un análisis exhaustivo de los requisitos y necesidades específicas de la institución educativa en cuestión. Posteriormente, se diseñará una arquitectura de software que garantice la integridad, seguridad y accesibilidad de los datos. La fase de desarrollo se centrará en la implementación de funcionalidades clave, asegurando una interfaz intuitiva y amigable para el usuario.

Este proyecto no solo representa un desafío académico en el campo de la ingeniería de software, sino también una oportunidad para contribuir de manera significativa al sector educativo, mejorando la eficiencia y calidad de los servicios ofrecidos por la institución. El resultado final no solo será un software de gestión de datos, sino una herramienta integral que potenciará el funcionamiento global de la institución educativa.

En resumen, el "Diseño y Desarrollo de un Software de Gestión de Datos para una Institución Educativa" aborda la necesidad imperante de modernizar los procesos administrativos en el ámbito educativo, destacando la importancia de la ingeniería de software como medio para alcanzar una gestión más eficaz y orientada a resultados.

3. Objetivo del Proyecto

3.1. Objetivo General

Involucrar a los estudiantes en un entorno real de trabajo para la ejecución de un proyecto de desarrollo de software educativo, donde apliquen los últimos métodos y técnicas de Ingeniería de Software que están permitiendo liberar productos tecnológicos de calidad.

3.2. Objetivos Específicos

1. Conocer las bases teóricas de Ingeniería de Software para la gestión de un proyecto de desarrollo de software.
2. Aplicar la Ingeniería de requerimientos para conocer las necesidades de los usuarios potenciales y establecer las personas comprometidas con el proyecto (stakeholder).
3. Comprender y realizar el proceso de análisis y diseño aplicando el modelo basado en prototipo para un software de gestión educativa.
4. Estudiar y comprender la arquitectura de un software de gestión educativa, documentando la estructura y funcionalidad de uno de sus módulos y desplegándolo en un servidor.
5. Dar a conocer el software de gestión educativa a través de actividades de dramatización y presentación a una audiencia.
6. Elaborar artículos científicos a partir de problemas resueltos con la creación de software.
7. Publicar el artículo en una revista indexada del grupo 2.

3.3. Actividades del proyecto de desarrollo de software (Ciclo de vida del software)

El proyecto de desarrollo de software ya tiene sus fases o actividades plenamente establecidas, gracias al Ciclo de vida del software ampliamente difundido en libros **sommerville11: 'softw'engin** y artículos; a continuación se explica como se han planificado actividades didácticas para que los estudiantes ejecuten estas fases para el desarrollo del software de gestión educativa.

1. Obtención de los requerimientos.

En esta fase del Ciclo de vida del desarrollo del software, se aplica el complejo proceso de ingeniería de requerimiento para conocer las necesidades funcionales y no funcionales que tienen

los usuario con respecto al sistemas informático a crear.

Al trabajar con estudiantes, a pesar de haberles brindado los conocimientos teóricos básicos para realizar este proceso, no se cuenta con la experiencia para abordar al usuario final, por lo que se utiliza la técnica de historia de usuario para preparar al estudiante en la obtención de requerimiento reales.

Para la creación de estas historias de usuario se organiza un taller Actividad A1-1 (4.2.1).

2. Análisis y Diseño.

En esta fase se realiza el análisis de las historias de usuario creada en la Actividad A1-1, y se utiliza el modelo basado en prototipos para crean los prototipos que son dibujos de como quedarían las páginas web que hacen de interfaces para que el sistema interactúe con el usuario; la creación de estos prototipos se la realiza en la Actividad A1-2 (4.4), B1-1 (4.4.1).

Para lograr que los estudiantes tuvieran las base cognitivas par el análisis y diseño, se activo la actividad B1-2, en el cual los estudiantes debieron revisar los vídeo y realizar comentario sobre lo aprendido.

3. Desarrollo o Implementación del sistema

Para esta fase se trabaja con un Sistemas de Información Web ya creado por el docente, el cual es un software libre y de código abierto lo que permite conocerlo en su totalidad, pero aunque el desarrollo no se tenga que realizar, para suplir esta actividad, se continua con la documentación perfeccionando los requerimientos y los prototipos de una manera más formal en la actividad A2-1 (4.8.1).

4. Despliegue.

Esta fase consiste en que ya contando con el sistema terminado, se lo ejecuta en el servidor y se lo coloca a disposición de los usuario para que ellos puedan hacer uso de el y al mismo tiempo probar su funcionalidad el cual debe coincidir con el trabajo realizado en la ingeniería de requerimiento, análisis y diseño.

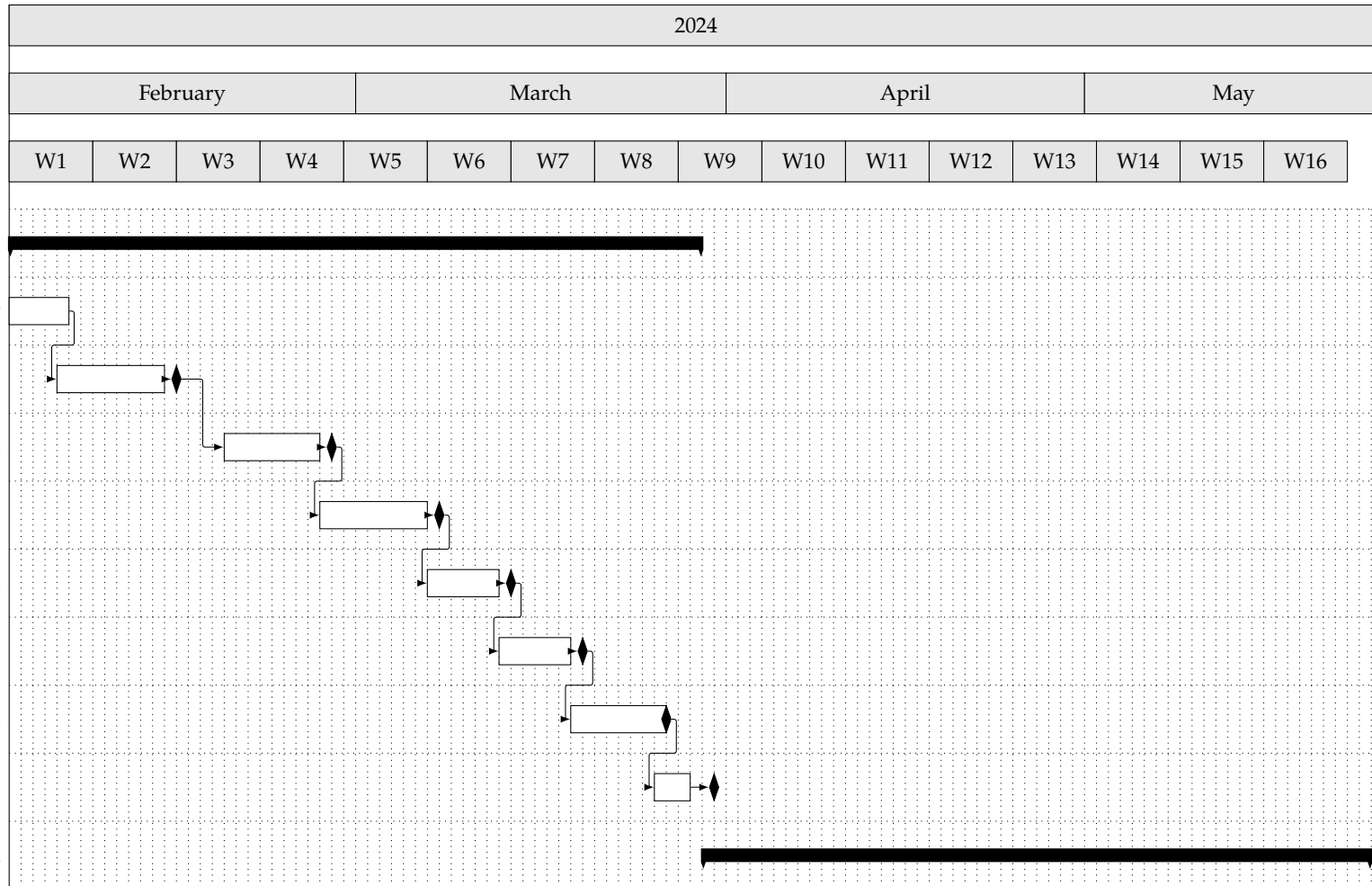
El aprendizaje y experiencia de esta fase se la logrará realizando la Actividad A2-2(4.9.1).

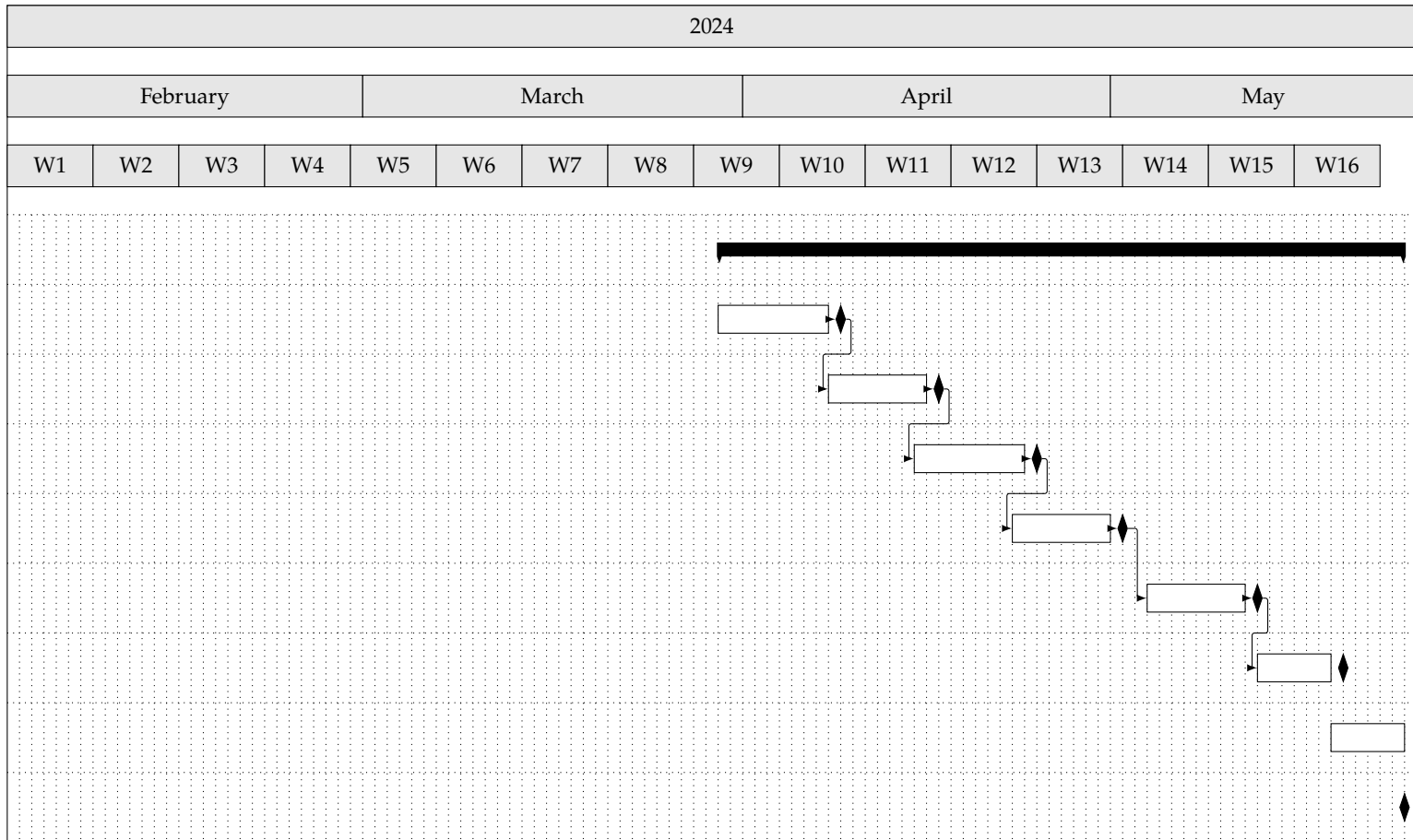
5. Mantenimiento y documentación.

A pesar de que el software esta creado, falta mucho por mejorar, en la fase de mantenimiento se busca esas oportunidades de mejoras, algo que se puede realizar gracias a que se tiene un pleno conocimiento de la arquitectura del software y las herramientas con que este fue elaborado.

El mantenimiento va de la mano con la documentación el cual los estudiantes lo realizaran en la Actividad B2-1(4.10.1) y la Actividad B2-2(4.10.2).

3.4. Diagrama de Gantt





7

4. Actividades de aprendizaje basado en Desarrollo del Proyecto

4.1. Actividades de configuración y aprendizaje base para el inicio del proyecto

Antes de comenzar la ejecución del proyecto, se debe ayudar a los estudiantes contar con los conocimientos básicos necesarios para lo cual se utiliza el material de Ingeniería de software video de Ingeniería de software.

- SE 1 : Introduction to Software Engineering
- SE 2 : Generic Process Model
- SE 3 : SDLC — Software Development Life Cycle With Real Life Example
- SE 4 : Waterfall Model in Software Engineering — Complete Explanation
- SE 8 : Prototyping Model
- SE 11: Agile Model
- SE 12: All SDLC Models Revisión
- SE 13: Funcional VS Non-Functional Requirements
- SE 14: Requirement Engineering — Establishing Ground Work
- SE 15: Requirement Engineering Tasks
- SE 16: Software Requirement Engineering Process
- SE 17: Software Requirement Specification (SRS)

En esta primera fase se conforma grupos de 5 estudiantes cada uno con uno de los siguientes roles:

Perfiles de los integrantes del grupo scrum:

- 1 Un estudiante de Product Owner.
- 2 Un estudiante de Scrum Master.
- 3 Tres estudiantes conformarán el equipo de desarrollador (Team developer).

Cada grupo escogerá un subsistema de los siguientes que conforman la plataforma web desarrollada:

1. Módulo de Registro y Login (Usuario: Docente)
2. Módulo de upload documentos.(Usuario: Director de carrera y docentes)
3. Módulo de Portafolios (Upload: docentes)
4. Módulo de inventario de artículo (usuario: bodegas)
5. Módulo de registro de visitas (usuario: secretaria de la carrera)

6. Módulo de Silabo (Usuario: docente).
7. Módulo de gestión de Asignaturas (Usuario: Director de carrera).
8. Módulo de Contabilidad Personal (Usuario: padres o madres).
9. Módulo de Evaluación (Usuario:Docentes).

Cada grupo recibirá una charla demostrativa de como funciona los módulos, y luego de conocerlos empezar la primera actividad.

4.2. Levantamiento de requerimientos

En esta fase del proyecto, 1ra fase del ciclo de desarrollo del software y del desarrollo de la asignatura Ingeniería de Software 1; los estudiantes deben prepararse para saber como recabar las necesidades del usuario sobre el sistemas de información que se le quiere construir, consciente de que se esta trabajando bajo una metodología Ágil de desarrollo llamada Scrum, una forma es creando historias de usuario.

4.2.1. Actividad A1-1

Taller de creación de historias de usuario

1 Objetivo de aprendizaje:

Esta actividad tiene como objetivo de aprendizaje principal, sumergir a los estudiantes en el proceso de ingeniería de requerimiento, aplicando la metodología Scrum, el grupo de estudiantes se pondrán en el zapato de los usuario para crear requerimientos funcionales y no funcionales del software deseado por los usuarios.

2 Instrucciones:

En el taller, cada grupo Scrum deberá crear un mínimo de 10 historias de usuario. Dado que cada equipo está conformado por 5 integrantes, se espera que cada miembro aporte al menos 2 historias de usuario. Todas las historias deberán estar alineadas con un mismo sistema, garantizando coherencia en el desarrollo del proyecto

3 Evaluación:

Cada estudiante subirá al MOODLE, el documento pdf con todas las historias de usuario creadas por el grupo.



Figura 1: Actividad A1-1:Taller para elaboración de Historias de usuario

Criterio	Excelente (4 puntos)	Bueno (3 puntos)	Aceptable (2 puntos)	Deficiente (1 punto)	Puntaje
Claridad	La historia es clara, precisa y comprensible sin ambigüedad	La historia es mayormente clara, con mínimas ambigüedades.	La historia es entendible pero tiene algunas ambigüedades.	La historia es confusa y difícil de interpretar.	
Formato correcto	Sigue perfectamente el formato: Como [rol], quiero [acción] para [beneficio]."	Sigue el formato, pero contiene leves errores.	El formato es reconocible pero incompleto o con errores significativos.	No sigue el formato estándar y es difícil identificar el propósito.	
Criterios de aceptación	Define criterios de aceptación completos, claros y verificables.	Define criterios de aceptación adecuados pero algo generales.	Los criterios de aceptación son vagos o incompletos.	No presenta criterios de aceptación o son irrelevantes.	
Alineación con el sistema	La historia está perfectamente alineada con los objetivos del sistema.	La historia está alineada, pero podría ser más precisa.	La alineación es parcial o poco relevante para el sistema.	La historia no tiene relación con el sistema propuesto.	
Trabajo en equipo	La historia refleja un trabajo colaborativo efectivo.	Hay evidencia de colaboración, aunque no uniforme.	El trabajo en equipo fue limitado y desigual.	No hubo evidencia de trabajo colaborativo.	

Cuadro 1: Rubrica de Evaluación: Historias de usuario

Escala de Puntuación

- 20 - 17 puntos: Excelente.
- 16 - 13 puntos: Bueno.
- 12 - 9 puntos: Aceptable.
- 8 o menos puntos: Deficiente.

Instrucciones para la evaluación:

1. **Total de historias:** Cada grupo debe presentar un mínimo de 10 historias de usuario, alineadas con un mismo sistema.
2. **Evaluación grupal:** La rúbrica será aplicada a cada historia, y luego se calculará un promedio para obtener la calificación grupal.
3. **Retroalimentación:** Se proporcionará feedback sobre las fortalezas y áreas de mejora.

4.3. Análisis y Diseño

En esta segunda etapa del Ciclo de Vida del Software, los estudiantes continúan realizando la Ingeniería de requerimiento, pero también empiezan con el Análisis de las historias de usuario para poder crear diseño de interfases, valiéndose del modelo de desarrollo de software basado en prototipos.

4.4. Actividad A1-2

Elaboración de prototipos

1 Objetivo de aprendizaje:

Esta actividad tiene como objetivo de aprendizaje principal, la comprensión del modelo basado en prototipo y el mejoramiento de las historias de usuario.

2 Instrucciones.

En esta actividad cada grupo scrum, elaborará los prototipos (dibujo de interfaces) 2 por cada integrante e hito los grupo presentarán las historias de usuario y los requerimientos funcionales y no funcionales extraídas de las historias de usuario .

3 Evaluación

Cada estudiante subirá al MOODLE, el documento con las 10 historias de usuario creadas en conjunto.



Figura 2: Actividad a1-2:Taller para la elaboración de prototipos

4.4.1. Actividad B1-1

Validar historias de usuario y prototipo

1 Objetivo de aprendizaje:

Esta actividad tiene como objetivo de aprendizaje principal, que los estudiantes puedan validar con el usuario real las historias de usuario y los prototipos realizados en el taller en clase.

2 Instrucciones.

En esta actividad los estudiantes mejoraran las historias de usuario y el prototipo realizados en la actividad anterior y lo validaran con el usuario real y potencial del sistema a desarrollar.

3 Evaluación:

La comprensión del artículo sera evaluado utilizando reactivos de alternativas multiples, esta evaluación sera con el artículo impreso en la mano, con la finalidad que el estudiantes puede encontrar las respuestas.



Figura 3: Actividad B1-1: Validación de historias de usuario y prototipos con usuario final del sistema.

4.4.2. Actividad B1-2

Escuchar y comentar los vídeos tutorales de la clase de Ingeniería de Software

1 Objetivo de aprendizaje:

Esta actividad tiene como objetivo de aprendizaje principal, recordar y comprender la fundamentos teóricos datos por el docente para el levantamiento de requerimiento, modelo basado en prototipo y metodología ágil para el desarrollo de software.

2 Instrucciones.

En este actividad los estudiante analizarán y comentara los vídeos subidos por el docente con la finalidad de contrastar con las actividades ya realizadas.

3 Evaluación

La comprensión del artículo sera evaluado utilizando reactivos de alternativas multiples, esta evaluación sera con el artículo impreso en la mano, con la finalidad que el estudiantes puede encontrar las respuestas.

The screenshot displays a YouTube interface for a video titled "Sesión No 12: Unidad 2: Modelo de desarrollo Ágil" uploaded by "stalin francis". The video has 187 views and 63 comments. The comments section shows several positive feedback messages from users like @mariaangelicavalenciabauti5070, @ximenahemicastillogonzal2171, @NeilCanchingre, and @mayerlyanapa6972.

Figura 4: Actividad B1-2: Ver, escuchar y comentar los vídeos de las sesiones clase

4.5. Actividad C1-1

Lectura de artículo científico

1 Objetivo de aprendizaje:

Esta actividad tiene como objetivo de aprendizaje principal, crear el hábito de lectura y análisis de artículos científicos sobre la tecnología de la información, también fortalecer la capacidad de aprendizaje autónomo y de forma específica comprender el uso y ventajas de los frameworks de programación basado en el patrón modelo-vista-controlador

2 Instrucciones.

En esta actividad el estudiante leerá de forma autónoma y comprensiva el artículo titulado “¿Existe una situación de crisis del software educativo?” e20: exist, este artículo debe leerlo varias veces a fin de comprender su contenido.

3 Evaluación

La comprensión del artículo será evaluado utilizando reactivos de alternativas múltiples, esta evaluación será con el artículo impreso en la mano, con la finalidad que el estudiante puede encontrar las respuestas.

¿Existe una situación de crisis del software educativo?

E. García Roselló, J. González Dacosta, M. Pérez Cota, J. B. García Pérez-Schofield,
V. G. Valdés Pardo

Resumen— A pesar de la importancia cada vez mayor de los computadores y el courseware en la educación, siguen existiendo serios problemas en el desarrollo de software educativo. A partir del análisis de esta situación, en el presente trabajo se plantea la existencia de una crisis del paradigma de desarrollo del software educativo. La solución más prometedora, si no la única, pasaría por un desarrollo completo de la ingeniería del courseware, especialmente de la parte de esta disciplina más cercana a la ingeniería del software educativo, es decir, a las metodologías aplicadas en el proceso de desarrollo de los sistemas software que forman parte del courseware.

Palabras Clave— Gestión del desarrollo de software, Ingeniería del software, Tecnología educacional.

I. PROBLEMAS ACTUALES DEL DESARROLLO DE SOFTWARE EDUCATIVO

El uso del computador como elemento integrante de procesos de enseñanza y aprendizaje se remonta a varias décadas atrás, y ha ido cobrando una creciente importancia, acentuada si cabe por la globalización de las comunicaciones y el acceso a la información proporcionado por la extensión de Internet y especialmente de la World Wide Web.

Así mismo, el desarrollo de *courseware* (entendido éste como una actividad formativa basada en o apoyada por computador) y de los sistemas basados en software educativo que lleva

de aplicaciones, librerías, frameworks, componentes y entornos de desarrollo de software educativo.

Pero a pesar de todo ello, y aunque ha habido multitud de propuestas metodológicas sobre el desarrollo de todo tipo de courseware, que de hecho han llevado al reconocimiento de la ingeniería del courseware como disciplina de pleno derecho [6], no ha habido un interés equiparable en el proceso de desarrollo del software con finalidad educativa que puede ser necesario construir como parte del courseware. Así, actualmente no se puede considerar que exista un conjunto de fundamentos, principios, métodos y herramientas claramente recogido, definido y ampliamente aceptado, que constituya la base de una completa ingeniería del software educativo.

Esto implica un claro vacío teórico y metodológico en la ingeniería del courseware. Por su propia naturaleza y por el tipo de productos que desarrolla, se trata de una ingeniería multidisciplinaria [6], donde un elemento significativo, a menudo el más importante, es el desarrollo de sistemas software.

Podemos concebir el núcleo primordial de la ingeniería del courseware como una fusión de otras dos disciplinas: el diseño instruccional por una parte, y la ingeniería del software por la otra, ya que, como es obvio pero a menudo no se valora suficientemente, el software educativo *es software* [6] [7] [8]. Pero mientras que la aplicación del diseño instruccional al proceso de desarrollo de courseware ha recibido una enorme atención, la parte más centrada en el desarrollo de los sistemas software que sustentan el courseware, y que correspondería a

Figura 5: Actividad C1-1:Lectura de artículo científico

4.6. Actividad C1-2

Lectura de artículo científico

1 Objetivo de aprendizaje:

Esta actividad tiene como objetivo de aprendizaje principal, crear el hábito de lectura y análisis de artículos científicos sobre la tecnología de la información, también fortalecer la capacidad de aprendizaje autónomo y de forma específica comprender la aplicación de la Metodología Ágil en el desarrollo de software.

2 Instrucciones.

En esta actividad el estudiante leerá de forma autónoma y comprensiva el artículo titulado “**Métodologías Ágiles en el Desarrollo de Software**” de José H. Canós, Patricio Letelier y M^a Carmen Penadés, este artículo debe leerlo varias veces a fin de comprender su contenido.

3 Evaluación

La comprensión del artículo será evaluado utilizando reactivos de alternativas múltiples, esta evaluación será con el artículo impreso en la mano, con la finalidad que el estudiante puede encontrar las respuestas.

Métodologías Ágiles en el Desarrollo de Software

José H. Canós, Patricio Letelier y M^a Carmen Penadés

DSIC -Universidad Politécnica de Valencia
Camino de Vera s/n, 46022 Valencia
{ jhcanos | letelier | mpenades }@dsic.upv.es

RESUMEN

El desarrollo de software no es una tarea fácil. Prueba de ello es que existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Por una parte tenemos aquellas propuestas más tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán. Estas propuestas han demostrado ser efectivas y necesarias en un gran número de proyectos, pero también han presentado problemas en otros muchos. Una posible mejora es incluir en los procesos de desarrollo más actividades, más artefactos y más restricciones, basándose en los puntos débiles detectados. Sin embargo, el resultado final sería un proceso de desarrollo más complejo que puede incluso limitar la propia habilidad del equipo para llevar a cabo el proyecto. Otra aproximación es centrarse en otras dimensiones, como por ejemplo el factor humano o el producto software. Esta es la filosofía de las metodologías ágiles, las cuales dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Este enfoque está mostrando su efectividad en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo pero manteniendo una alta calidad. Las metodologías ágiles están revolucionando la manera de producir software, y a la vez generando un amplio debate entre sus seguidores y quienes por escepticismo o convencimiento no las ven como alternativa para las metodologías tradicionales. En este trabajo se presenta resumidamente el contexto en el que surgen las metodologías ágiles, sus valores, principios y comparación con las metodologías tradicionales. Además se describen brevemente las principales propuestas, especialmente Programación Extrema (eXtreme Programming, XP) la metodología ágil más popular en la actualidad.

PALABRAS CLAVE. Procesos de Software, Metodologías Ágiles, Programación Extrema (XP)

1. INTRODUCCIÓN

En las dos últimas décadas las notaciones de modelado y posteriormente las herramientas pretendieron ser las “balas de plata” para el éxito en el desarrollo de software, sin embargo, las expectativas no fueron satisfechas. Esto se debe en gran parte a que otro importante elemento, la metodología de desarrollo, había sido postergado. De nada sirven buenas notaciones y herramientas si no se proveen directivas para su aplicación. Así, esta década ha comenzado con un creciente interés en metodologías de desarrollo. Hasta hace poco el proceso de desarrollo llevaba asociada un marcado énfasis en el control del proceso mediante una rigurosa definición de roles, actividades y artefactos, incluyendo modelado y documentación detallada. Este

Figura 6: Actividad C1-2: Lectura de artículo científico

4.7. Desarrollo del software aplicando la metodología Ágil - Scrum

Bajo el concepto de la metodología Ágil (Scrum) se comienza al desarrollar o hacer mejoras al software.

4.7.1. Actividad E1

Dramatización del proceso de desarrollo usando Scrum

1 Objetivo de aprendizaje:

Sistematizar las actividades realizadas por cada uno de los integrantes del equipo Scrum, mediando la acción de dramatización del personaje.

2 Instrucciones.

En esta actividad cada grupo de estudiantes creado al inicio de clase scrum, realizará una dramatización del trabajo realizado siguiendo la metodología Scrum par el desarrollo del software.

3 Evaluación

La comprensión del artículo sera evaluado utilizando reactivos de alternativas multiples, esta evaluación sera con el artículo impreso en la mano, con la finalidad que el estudiantes puede encontrar las respuestas.



Figura 7: Dramatización Actividad E1

4.8. Análisis y diseño

4.8.1. Actividad A2-1

Documentos de especificaciones de requerimientos:

- 1 Para desarrollar esta actividad utilizar el formato de Documentos Especificación de Requerimiento. .
- 2 Para acelerar la edición de esta documento utilizar Chatgpt a fin de crear el documento base.
- 3 Leer y corregir el documento generado adaptándolo de forma más precisa al contexto.
- 4 Subir el documento con el nombre de los integrantes a la plataforma MOODLE.

Documento de Especificación de Requerimientos

Versión	1.0 Inicial Básica
Fecha	Marzo 15 de 2008
Autor	Grupo Los 5 Magníficos

Contexto

La librería de la Universidad *El Pensador* quiere ofrecer a sus usuarios una tienda virtual para sus productos, libros, ropa, cuadernos, esferos, etc.. Quiere contar con un catálogo de la librería en línea donde se pueda hacer búsquedas bajo distintos criterios y adquirir los productos por internet.

Alcance

Este sistema se ocupará solo de la venta de los libros.

Glosario de Términos

Glosario

Concepto	Descripción
Book	Libro en la tienda.
Author	Autor de uno o muchos libros disponibles en la tienda. Un libro puede tener varios autores.
Editorial	representa las empresas Editoriales de los libros. Cada libro tiene una editorial.

Requerimientos Funcionales

Actores

Para este sistema hay dos actores: el usuario *Comprador* y el *Administrador*.
El *Administrador* es quien se ocupará de crear la información en el sistema.

Casos de Uso

	Nombre	Resumen
CU1	Consultar datos de un libro	El sistema permite que un usuario consulte la información de un libro en particular
CU2	Crear un libro	El sistema permite el registro de un libro nuevo en la tienda
CU2	Ver la información de todos los libros	El sistema muestra el catálogo de libros de la librería.

Reglas de negocio

- No existen dos libros con el mismo ISBN
- No se puede eliminar un autor si tiene algún libro registrado

Requerimientos No Funcionales

Figura 8: Actividad A2-1: Documento Especificaciones de Requerimiento

4.9. Despliegue

En esta etapa del Ciclo de Vida del Software, el software ya está creado y listo para ser presentado y utilizado por el usuario final, a esta disponibilidad del software se le llama despliegue.

4.9.1. Actividad A2-2

Despliegue de módulo:

Objetivo de aprendizaje El estudiante será capaz de reconocer los archivos modelos-vistas-controlador de un modelo que compone un sistema de información web, además realizará el despliegue del módulo en un servidor de pago.

Instrucciones: A cada grupo de desarrollo scrum se le asignará uno de los siguientes módulos, el cual lo estudiarán y lo desplegarán en un servidor siguiendo las instrucciones del docente.

- | | | |
|----------------|----------------|------------------|
| 1 Sexo | 1 Estado civil | 1 Tipodocumento |
| 2 Tipo sangre. | 2 Tipo sangre. | 2 Tipoevento. |
| 3 Pais. | 3 Pais. | 3 tipocalidad. |
| 4 Ciudad. | 4 Ciudad. | 4 tipomatricula. |

Evaluación: La evaluación consistirá simplemente en ver que el módulo se ejecute de forma correcta en el servidor.

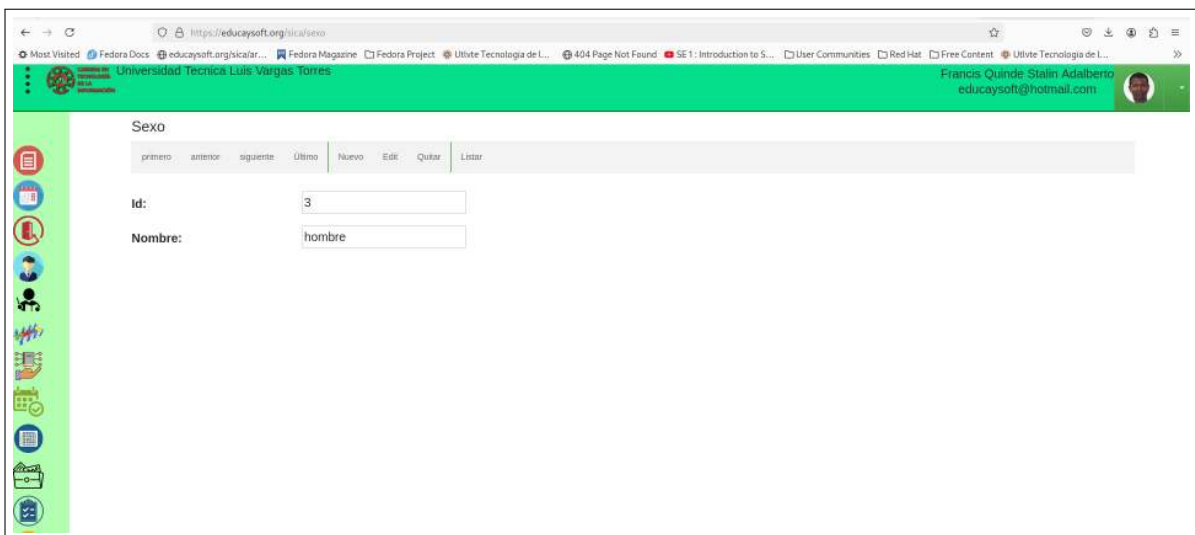


Figura 9: Actividad A2-2: Despliegue de módulo

4.10. Documentación, mantenimiento y difusión

En las siguientes actividades los estudiantes Documentarán, darán mantenimiento y difundirán el proyecto de desarrollo de software.

4.10.1. Actividad B2-1

Documentación del módulo desplegado

1 Objetivo de aprendizaje:

Con esta actividad los estudiantes serán capaces de documentar de forma técnica y comprensible la estructura del módulo por ellos desplegado

2 Instrucciones.

El grupo creará un documento compartido y documentarán el módulo según un formato dado por el docente en documento guía.

3 Evaluación

Se evaluará el documento el cual debe estar completo y detallado.

En el diagrama de la figura 3, el objeto principal persona se identifica a los atributos compuestos aquellos que no pueden estar dentro de la tabla principal por lo que se crea otra tabla que contiene este atributo pero mantiene una relación ya sea uno a muchos o muchos a muchos con la tabla persona. En la figura 4, se muestra otro diagrama entidad relación de la tabla persona pero esta vez la relación la tiene con tabla que representa las instancias que esta tabla persona puede tener como objeto, lo que ayuda principalmente a no duplicar atributos, manteniendo los atributos de las instancias centralizada en la tabla persona, esto también ayuda a evitar atributos vacíos.

Para un mejor entendimiento, se puede decir que una persona puede tener varios roles entre estos están que el puede ser usuario, estudiante, docente, funcionario, donde cada uno de estos puede ser repetido para una sola persona, un caso práctico es el de usuario, donde una misma persona puede tener varios usuarios.

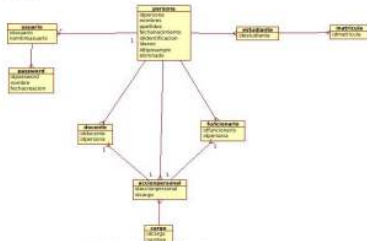


Figura 4: Diagrama entidad relación con las instancias de la tabla persona

Fuente: Elaboración propia.

29

4.6. Descripción de las tablas a crear

En la siguiente tabla se describen cada una de las tablas que conforman el sistema.

Tabla 22: Descripción de las tablas de la base de datos.

No	Nombre de tabla	Descripción
1	persona	Tabla donde se almacena los datos personales.
2	usuario	Tabla donde se almacena los datos del usuario
3	Módulo	Tabla donde se almacena todos los módulos que conforman el sistema.
4	acceso	Tabla que almacena los módulos a los cuales tiene acceso un usuario
5	Artículo	Tabla donde se almacena la descripción de un artículo existente en la institución.
6	Ubicacionarticulo	Tabla que contiene la información de donde está ubicado el artículo dentro de la institución y quien es el custodio
7	Custodio	Tabla que contiene la relación entre persona y artículo para saber quien está a cargo del artículo.
8	Prestamoarticulo	Tabla que contiene la información de quien presta un artículo en que fecha y a que hora, así como su devolución.
9	Documento	Tabla donde se almacena los metadatos de un

Figura 10: Diagrama entidad relación

Fuente: Elaboración propia.

4.10. Diagrama De Clases Login

El diagrama de clases presentado describe la estructura y relaciones entre las diferentes clases e interfaces involucradas en el proceso de login de la aplicación. A continuación, se explica cómo

32

se relacionan las diferentes entidades y cómo funcionan en conjunto para realizar el proceso de autenticación y gestión de usuarios.

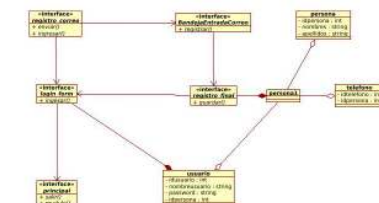


Figura 5: Diagrama de clase proceso acceso al sistema

Fuente: Elaboración propia.

En este diagrama de clase la actividad comienza con presentar una interfaz para que el nuevo usuario que quiere tener acceso al sistema, registre su correo electrónico, esta interfaz implementa los métodos enviar() e ingresar(), al escribir su correo electrónico y ejecutar el método enviar(), un mensaje es enviado a la bandeja de entrada del correo registrado. El usuario debe acceder a su correo electrónico y desde ahí revisar el mensaje recibido por el sistema, en este mensaje existe un link, el cual le permite tener acceso a la interfaz registro_final, esta interfaz es un formulario que esta relacionado por una clase persona1, esta clase carga los datos de varias tablas de la base de datos: entre estas la tabla persona, en este formulario el usuario ingreso sus datos de contacto solicitados, esta interfaz implementa el método guardar(), el cual al ser ejecutado dispara la interfaz login_form, que le permite al usuario ya ingresar sus credenciales registradas (usuario, contraseña) las cuales son validas y luego carga los datos del usuario, finalmente después de verificar que las credenciales son

33

correctas, entonces presenta la interfaz principal que corresponde el menú que habilita los módulos para poder trabajar dentro del sistema.

Figura 11: Diagrama de Clases

4.10.2. Actividad B2-2

Video explicativo de la arquitectura del computador y como estan compuesto

Explicación del módulo

1 Objetivo de aprendizaje:

Los estudiantes serán capaz de explicar de forma técnica y clara la estructura y funcionamiento de módulo desplegado.

2 Instrucciones.

Cada grupo preparara un dispositiva y entre todos harán la exposición mediante un vídeo que luego sera editado según las instrucciones del docente.

3 Evaluación

Se evaluara según una rubrica.

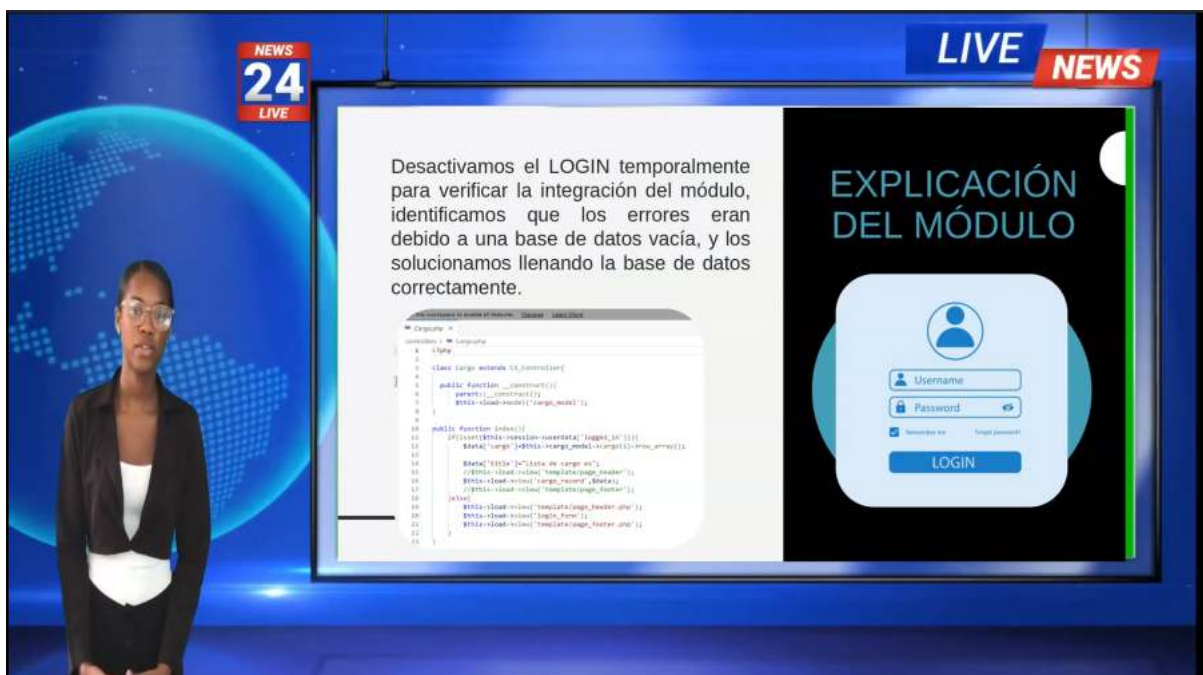


Figura 12: Actividad B2-2: Explicación de modulo desplegado

4.10.3. Actividad C2-1

Lectura de artículo científico

1 Objetivo de aprendizaje:

Esta actividad tiene como objetivo de aprendizaje principal, crear el hábito de lectura y análisis de artículos científicos sobre la tecnología de la información, también fortalecer la capacidad de aprendizaje autónomo y de forma específica comprender el uso y ventajas de los frameworks de programación basado en el patrón modelo-vista-controlador

2 Instrucciones.

En esta actividad el estudiante leerá de forma autónoma y comprensiva el artículo titulado **“Frameworks PHP basados en la arquitectura Modelo-Vista-Controlador para desarrollo de aplicaciones web”** carlos22: 'framework' 'php' 'model' 'vista' 'contr', este artículo debe leerlo varias veces a fin de comprender su contenido.

3 Evaluación

La comprensión del artículo será evaluado utilizando reactivos de alternativas múltiples, esta evaluación será con el artículo impreso en la mano, con la finalidad que el estudiantes puede encontrar las respuestas.

Artículo de revisión

Frameworks PHP basados en la arquitectura Modelo-Vista-Controlador para desarrollo de aplicaciones web

PHP frameworks based on the Model-View-Controller architecture for web application development

Carlos Andrés Castillo Yagual
Marjorie Alexandra Coronel Suárez

Universidad Estatal Península de Santa Elena | La Libertad - Ecuador | CP 240204

ccastilloy@upse.edu.ec

<https://orcid.org/0000-0001-6578-0480>
<https://orcid.org/0000-0003-3543-5357>

<https://doi.org/10.26423/rctu.v10i11.703>
Páginas: 70-78

Resumen

En este trabajo se analizan los frameworks PHP de código abierto basados en el patrón de diseño Modelo-Vista-Controlador (MVC) a través de cuadros comparativos, que permiten a los desarrolladores crear sus aplicaciones web sin necesidad de codificar desde cero, ya que cuentan con herramientas integradas y bibliotecas que facilitan las operaciones CRUD (creación, lectura, actualización y eliminación), organización del código, facilidad de mantenimiento, seguridad ante ataques de inyección SQL y otras amenazas. Se describen las características, especificaciones técnicas, componentes de los frameworks considerados los más comunes como son: Laravel, Symfony, CodeIgniter, Zend, CakePHP, Yii, que ayudan a reducir el tiempo en el proceso de desarrollo empleando buenas prácticas de programación. Para el estudio se aplicó el método analítico-sintético para identificar los frameworks que permitan diseñar e implementar web dinámicas, robustas y seguras facilitando la corrección de errores. Los resultados brindan un insumo a los programadores

Abstract

This report analyses the open-source PHP frameworks based on the Model-View-Controller (MVC) design pattern through comparative tables, which allow developers to create their web applications without the need to code from scratch, as they have integrated tools and libraries that facilitate CRUD operations (creation, reading, updating, and deletion), code organization, ease of maintenance, security against SQL injection attacks and other threats. The characteristics, technical specifications, and components of the most common frameworks are described, such as Laravel, Symfony, CodeIgniter, Zend, CakePHP, and Yii, which help reduce development time by employing good programming practices. For the study, the analytical-synthetic method was applied to identify frameworks that allow the design and implementation of dynamic, robust, and secure websites, facilitating the correction of errors. The results provide input to programmers and software engineers who are dabbling in PHP

Figura 13: Actividad C2-1: Aprendizaje autónomo - Lectura de artículo científico.

4.10.4. Actividad C2-2

Lectura de artículo científico

1 Objetivo de aprendizaje:

Esta actividad tiene como objetivo de aprendizaje principal, crear el hábito de lectura y análisis de artículos científicos sobre la tecnología de la información, también fortalecer la capacidad de aprendizaje autónomo y de forma específica comprender el uso y ventajas de los frameworks de programación basados en el patrón modelo-vista-controlador

2 Instrucciones.

En esta actividad el estudiante leerá de forma autónoma y comprensiva el artículo titulado "DevOps in Practice for Education Management Information System at ECNU" dawei20:'devop'pract'educat'manag'infor'system'ecnu, este artículo debe leerlo varias veces a fin de comprender su contenido.

3 Evaluación

La comprensión del artículo será evaluado utilizando reactivos de alternativas múltiples, esta evaluación será con el artículo impreso en la mano, con la finalidad que el estudiante puede encontrar las respuestas.

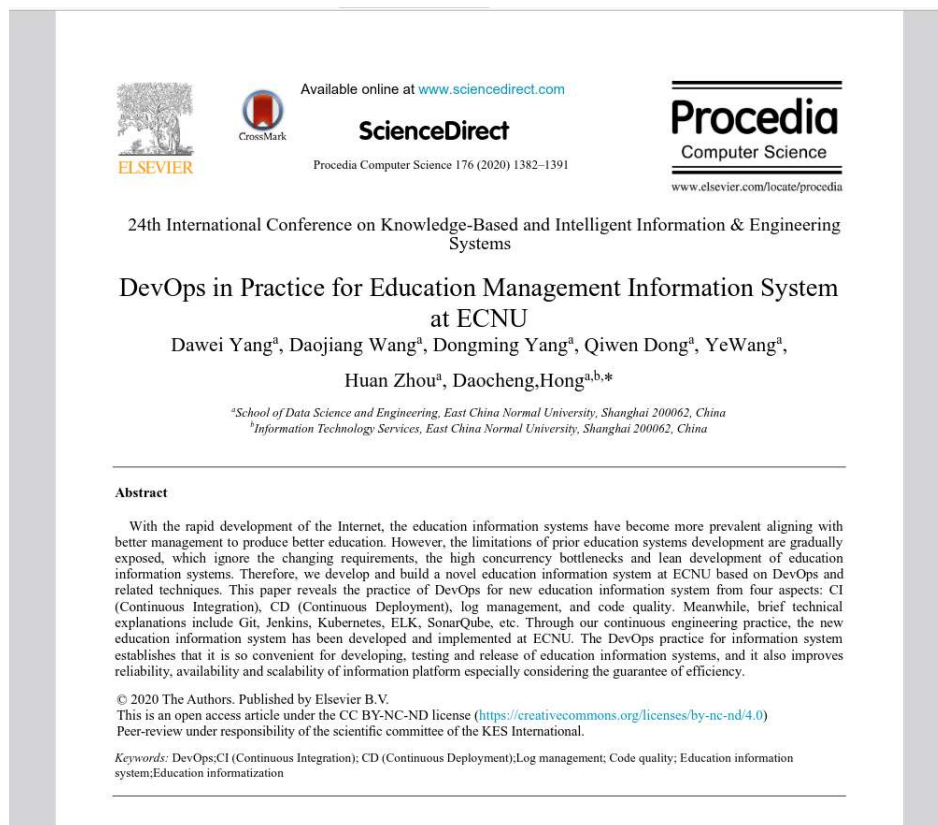


Figura 14: Actividad C2-2: Aprendizaje autónomo Lectura de artículo científico.

4.11. Actividad E2

Presentación de los módulos del sistema sica:

Objetivo de aprendizaje: Los estudiante son capaces de presentar el funcionamiento y la estructura de un modulo complejo de un sistemas de información después de haberlo desarrollado aplicando la ingeniería de Software.

Instrucciones En un evento en el salón Bernal y vestidos de manera formal, los estudiante presentaran los módulo más completos de sistema de información Sica.

1 Persona	6 Distributivo	11 Evento
2 Documento	7 Docente	12 Silabo
3 Teléfonos	8 Asignatura	13 Temas
4 Correos	9 Distributivo_docente	14 Calendario_Académico.
5 Portafolio_digital	10 Asignatura_Docente	15 Certificados

Evaluación: Se evalúa el material presentado y la presentación de cada integrante del grupo.



Figura 15: Actividad E2:Presentación final del producto del Proyecto de desarrollo de Software

5. Resultados

1. Los estudiantes a través de una metodología activa de aprendizaje comprenden la primera etapa del desarrollo del software que consiste en el levantamiento de los requerimientos.
2. Utilizando la dramatización interiorizan las actividades a desarrollar en la metodología Scrum para la ejecución del proyecto de desarrollo de software.
3. Los estudiantes a través de exposiciones comprenden la arquitectura del software.
4. Los estudiantes comprenden la importancia de las lecturas de artículos científicos, para el desarrollo de software.

6. Conclusiones

1. La metodología ágil, crea un marco de trabajo ideal no solo para entornos de desarrollo, sino que también para entornos educativos ya que justifica la creación de equipos de trabajo.
2. La creación previa de un software bajo la modalidad de software libre se convirtió en una herramienta fundamental en el proceso de aprendizaje y una fuente de motivación para los estudiantes.
3. La metodología de aprendizaje activo comenzando desde la metodología basada en proyectos, complementada con actividades de dramatización de modelos Scrum, y eventos de presentación de software fue fundamental para la comprensión de la asignatura de ingeniería de software.¹